



**Ministerio de Cultura y Educación**  
**Universidad Nacional de San Luis**  
**Facultad de Ciencias Físico Matemáticas y Naturales**  
**Departamento: Informatica**  
**Area: Area IV: Pr. y Met. de Des. del Soft.**

**(Programa del año 2016)**

### I - Oferta Académica

Materia	Carrera	Plan	Año	Período
INGENIERIA DEL SOFTWARE	PROF.CS.COMPUT.	02/16	2016	1° cuatrimestre
INGENIERIA DE SOFTWARE I	LIC.CS.COMP.	32/12	2016	1° cuatrimestre
INGENIERIA DE SOFTWARE I	PROF.CS.COMPUT.	06/08	2016	1° cuatrimestre

### II - Equipo Docente

Docente	Función	Cargo	Dedicación
ABDELAHAD, CORINA NATALIA	Prof. Responsable	P.Adj Exc	40 Hs
BERNARDIS, EDGARDO	Auxiliar de Práctico	A.1ra Simp	10 Hs
PEREZ, NORMA BEATRIZ	Auxiliar de Práctico	A.1ra Semi	20 Hs

### III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	Teóricas	Prácticas de Aula	Práct. de lab/ camp/ Resid/ PIP, etc.	Total
Hs	3 Hs	3 Hs	2 Hs	8 Hs

Tipificación	Periodo
B - Teoria con prácticas de aula y laboratorio	1° Cuatrimestre

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
14/03/2016	24/06/2016	15	120

### IV - Fundamentación

Se introduce al alumno en el desarrollo de software como una actividad ingenieril y al trabajo en equipo en el contexto de la Teoría General de Sistemas.

Se pretende que el alumno aprenda los fundamentos básicos de cada una de las etapas que comprende el proceso de desarrollo de software así como las distintas técnicas y metodologías aplicables, conocimientos de los principios de diseño así como de las principales técnicas de validación y verificación del software.

### V - Objetivos / Resultados de Aprendizaje

Al finalizar esta asignatura se espera:

- Preparar al alumno para su desempeño en la Industria del Software con un enfoque Sistémico.
- Que el alumno asimile los conceptos de procesos de desarrollo de software, desde su inicio, análisis, diseño hasta su verificación y validación.
- Que el alumno tome conocimiento de conceptos básicos de ingeniería de software desde un punto de vista de la automatización de procesos de desarrollo de software.

Para cubrir dichos objetivos se integrarán conceptos, modelos y métodos en un proyecto integrador.

### VI - Contenidos

**Teoría General de los Sistemas. Finalidad, Naturaleza y Características de los Sistemas de Información. El Producto**

**de Software integrado en un Sistema de Información. El proceso de software: Ciclos de vida, herramientas. Ingeniería de requerimientos, introducción a los métodos formales. Modelado Estático y Modelado Dinámico. Análisis, diseño, implementación, verificación, validación y mantenimiento de software. Diseño Centrado en el Usuario. Conceptos, Herramientas y Estándares de Calidad de Software del Producto y del Proceso. Privacidad, Integridad y Seguridad de Sistemas de Software.**

## **PROGRAMA:**

### **Unidad I:**

**Del enfoque Analítico-Reduccionista a la Teoría General de los Sistemas. El Enfoque Sistémico. Concepto de Sistema.**

Características de los Sistemas. Elementos constitutivos de un Sistema. Clasificación de los Sistemas. Sistemas de Información. Elementos constitutivos de un Sistema de Información. Criterios de clasificación de los Sistemas de Información. Ejemplos de distintos tipos de Sistemas de Información.

### **Unidad II:**

**Ingeniería de Software. Dominio de la IS. Aspectos fundamentales. Características del software. Procesos, metodologías y herramientas de software. Modelos de procesos de desarrollo de software. Evolución de las metodologías de desarrollo de sistemas de software. Ingeniería de Requerimientos. Ingeniería de la Información.**

### **Unidad III:**

**Modelado en UML. Introducción. Modelos y diagramas. Importancia de los modelos. Diagramas estructurales. Clases: atributos, operaciones y responsabilidades. Relaciones: dependencia, generalización y asociación. Diagramas de Clases y Diagramas de Objetos. Diagramas de comportamiento. Diagramas de Secuencia y Comunicación. Diagramas de Casos de Usos.**

### **Unidad IV:**

**Proceso de Desarrollo. Análisis, diseño, implementación, pruebas y mantenimiento de software. Modelos del Análisis. Modelo de Casos de Usos del sistema. Modelo conceptual o del dominio. Modelos del Diseño.**

### **Unidad V:**

**Metodologías Ágiles. Conceptos. Principios. Manifiesto. Ventajas y desventajas de las metodologías ágiles. Estudio de utilización concreta de las metodologías ágiles en la industria del software: Extreme Programming.**

### **Unidad VI:**

**Validación y Verificación. Pruebas de errores del software. Estrategias de prueba. Técnicas de prueba: prueba estructural y prueba funcional. Prueba Unitaria. Uso de JUnit. Calidad del Software. Aseguramiento de la calidad del**

**software. Medidas de fiabilidad y disponibilidad. Gestión de Configuración del Software. Conceptos. Control. Garantía. Costos. Líneas base, elementos y objetos en la configuración del software. Versiones.**

#### **Unidad VII:**

**Privacidad, integridad y seguridad en Sistemas de Información. Privacidad en Internet. Integridad: Conceptos y herramientas básicas para preservarla. Integridad en el caso de aplicaciones distribuidas.**

#### **Unidad VIII:**

**Métodos Formales. Conceptos básicos. Deficiencias de los enfoques menos formales. Las matemáticas en el desarrollo de software. Notación matemática para la especificación formal. Lenguajes de especificación formal. Especificaciones formales en Alloy.**

### **VII - Plan de Trabajos Prácticos**

Practico 1: La Ingeniería de Software y los modelos de ciclo de vida del software.

Práctico 2 - parte I: UML: Diagramas de clase y de objetos

Práctico 2 - parte II: UML: Diagramas de casos de uso

Práctico 2 - parte III: UML: Diagramas de secuencia y comunicación.

Práctico 3: Prueba estructural.

Práctico 4: Prueba funcional.

Práctico 5: Prueba unitaria con JUnit.

Práctico 6: Especificaciones formales del software.

Laboratorio integrador: Se solicita la ejecución de un proyecto para llevar adelante un caso de estudio donde se cubran los aspectos fundamentales de la Ingeniería de Software, empleando herramientas automatizadas que den soporte a las diversas actividades llevadas a cabo durante el proceso de desarrollo.

### **VIII - Regimen de Aprobación**

La materia se desarrolla con la modalidad de promoción sin examen final.

Existen dos niveles:

a) Regularización solamente: Para regularizar la materia se deberá:

1.- Tener como mínimo un 70% de asistencia a clases.

2.- Presentación y aprobación del proyecto integrador de laboratorio con nota mayor o igual a 6 (seis).

3.- Aprobar dos parciales con una nota mayor o igual a 6 (seis).

Cada parcial posee dos recuperaciones. En caso de no aprobar un parcial, deberá aprobar alguno de sus recuperatorios con nota mayor o igual a 6 (seis).

b) Promoción sin examen final: Para regularizar y aprobar la materia se deberá:

1.- Tener como mínimo un 80% de asistencia a clases.

2.- Presentación y aprobación del proyecto integrador de laboratorio con nota mayor o igual a 7 (siete).

3.- Aprobar dos parciales o sus respectivas recuperaciones con una nota mayor o igual a 7 (siete). Cada parcial posee dos recuperaciones. En caso de no aprobar un parcial, deberá aprobar alguno de sus recuperatorios con nota mayor o igual a 7 (siete).

4.- Aprobar una evaluación final integradora con una nota mayor o igual a 7 (siete).

Aquellos alumnos que sólo regularicen la materia deberán rendir un examen final, en los turnos establecidos.  
Alumnos Libres: Por las características propias del proyecto de laboratorio a desarrollarse durante todo el cuatrimestre, no se aceptan alumnos libres.

## **IX - Bibliografía Básica**

- [1] J. O' Brien & G. Marakas, "Introduction to Information Systems", 15 th edition.
- [2] Pressman, Roger; "Ingeniería del Software - Un enfoque práctico", Mc Graw Hill, 6ta Edición. 2006.
- [3] Pankaj Jalote; An Integrated Approach to Software Engineering, 3ra edición. Springer. 2005.
- [4] Booch, Rumbaugh, Jacobson. The Unified Modeling Language User Guide, 2nd Edition. Addison-Wesley, 2005
- [5] Booch, Rumbaugh, Jacobson. The Unified Modeling Language Reference Manual, 2nd Edition. Addison-Wesley, 2005.
- [6] Ghezzi, Carlo y otros "Fundamentals of Software Engineering", Prentice Hall, 1991.
- [7] Jorgensen, Paul C.; "Software Testing, a Craftsman's Approach", CRC Press, 1995.
- [8] Daniel Bolaños y otros; "Pruebas de software y JUnit", Prentice-Hall, 2008.
- [9] Martín Fowler, The New Methodology, <http://www.martinfowler.com/articles/newMethodology.html>
- [10] Kent Beck, "Extreme Programming Explained", 1ra edición, 1999.
- [11] Scott Ambler, "Agile Modeling: Effective practices for Extreme Programming and the Unified Process", John Wiley & Sons, 2002.
- [12] Daniel Jackson; "Software Abstractions", The MIT Press, 2012.
- [13] Apuntes de la Cátedra.

## **X - Bibliografía Complementaria**

- [1] Booch, Grady. Object-Oriented analysis and design with applications. The Benjamin/Cummings Publishing Company Inc. 1994.
- [2] Stephen Schach. "Ingeniería de Software Clásica y Orientada a Objetos, 6ta ed.", Mc Graw Hill, 2005.
- [3] Sommerville, Ian; "Software Engineering", Addison-Wesley, 6ta Edición. 2002.
- [4] Beck, Kent. Test-driven development by example. Pearson Education, 2003.
- [5] Scott Ambler, "Agile Modeling and the Unified Process", <http://www.agilemodeling.com/essays/agileModelingRUP.htm>, 2002.
- [6] Pekka Abrahamsson, Outi Salo, Jussi Ronkainen & Juhani Warsta, "Agile Software Development Methods: Review and Analysis", VTT, 2002.
- [7] "Extreme Programming: A gentle introduction", <http://www.extremeprogramming.org/>
- [8] Página Web del Instituto de Ingeniería de Software (CMU). <http://www.cmu.edu>,

## **XI - Resumen de Objetivos**

Asimilar los conceptos de procesos de desarrollo de software, desde su especificación, análisis, diseño hasta su verificación y validación, incorporando conceptos de privacidad, seguridad y calidad del software.

## **XII - Resumen del Programa**

Teoría General de los Sistemas. Sistemas de Información. El Producto de Software integrado en un Sistema de Información. El proceso de software: Ciclos de vida, herramientas. Ingeniería de requerimientos, introducción a los métodos formales. Análisis, diseño, implementación, verificación, validación y mantenimiento de software. Conceptos de calidad de software. Conceptos de privacidad, integridad y seguridad en Sistemas de Información.

## **XIII - Imprevistos**

## **XIV - Otros**